This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Problem Image Mailbox.

Ref T14996A

A VHSIC ETM-BUS–COMPATIBLE TEST AND MAINTENANCE INTERFACE

LaNae Avra

Honeywell, Inc.
Solid State Electronics Division
12001 Highway 55
Plymouth, MN 55441

ABSTRACT

This paper describes an on-chip Test and Maintenance Control (TMC) function block which provides an interface to the VHSIC standard Element Test and Maintenance bus (ETM-bus). Using commands received serially over the ETM-bus, the TMC controls chip-level test and maintenance features such as chip initialization, serial scan, debug, and built-in self-test operations. The TMC also monitors chip functional operation, providing status information to the ETM-bus controller upon request, and sending an interrupt to the controller when it detects error conditions on the chip.

The test and maintenance hierarchy in which the TMC is implemented is described, with an emphasis on the operation of the ETM-bus. The TMC is then described in terms of its architecture, the operation of each of its function blocks, and how it controls chip operation. Finally, TMC enhancements and variations are suggested.

INTRODUCTION

The increased complexity and speed of today's advanced integrated circuits require a greater understanding of, and respect for, the problems associated with circuit testability and maintainability. The incorporation of testability-improvement techniques such as serial access to internal storage devices (serial scan),[1,2,3] serial access to primary chip inputs and outputs (boundary scan),[4,5,6] and built-in self-test techniques[7,8,9] during the early stages of the design process (known as design-for-test) is becoming increasingly popular among testability-conscious digital VLSI designers. Efficient control of, and access to, this design-for-test (DFT) circuitry necessitates the inclusion of a standard, on-chip test interface.

A test interface consisting of a standard chip-to-chip communication path and standard on-chip test interface circuitry allows for automated test and monitoring of multi-chip modules by a single controller during debug, manufacturing, and field operation. In order to merit widespread use and gain acceptance from VLSI designers, the test interface circuitry must not unduly impact chip performance or exceed hardware overhead constraints (i.e., pin, power, and size). In addition, the test interface circuitry must be flexible to accommodate a variety of on-chip DFT techniques, and its architecture and operation must have a minimum impact on the design and test cycle-times.

One test interface design that has been proposed[10] encourages the extensive use of test points[11] (visibility points and controllability points) in the application circuitry. While an on-chip test interface for VLSI devices should provide for the control and monitoring of internal test points, it should also provide built-in self-test capabilities and should encourage the use of high-fault-coverage techniques such as serial scan. The number of ad hoc test points which must be added to provide sufficient fault coverage for VLSI chips represents an unreasonable application/test interface interconnect overhead.

Another recent on-chip test interface design is the testprocessor described by Blaschke, et al.,[12] which incorporates many of the control and test pattern storage capabilities that are becoming necessary as more and more self-test and monitoring processes are performed on-chip. However, the testprocessor is intended for implementation on integrated circuits of the future and would be impractical as a function block on VLSI devices with less than 1 million transistors.

In order to meet today's VLSI test and maintenance requirements while representing acceptable performance impact, hardware overhead, and design and test cycle-time impact, the Honeywell Solid State Electronics Division has developed an on-chip Test and Maintenance Control (TMC) function block which forms the basis of a structured testability hierarchy that reduces system test complexity to a manageable level. The TMC interfaces with the VHSIC (Very High Speed Integrated Circuits) standard Element Test and Maintenance bus (ETM-bus) to control on-chip test and maintenance features and DFT circuitry. The TMC receives instructions and test data over the serial ETM-bus and uses this information to control serial scan operations, chip debug, built-in self-test, and error monitoring and reporting. The TMC, which has been implemented on several Honeywell VLSI chip designs, has a structured architecture that can be easily modified to control a variety of DFT circuitry while remaining compatible with the standard ETM-bus protocol.

THE TEST AND MAINTENANCE HIERARCHY

The Test and Maintenance (T&M) hierarchy in which the TMC is implemented makes use of a two-level test bus structure that was developed by Honeywell, IBM, and TRW under the direction of the Department of Defense (DoD) to achieve the multi-vendor interoperability goals and requirements of the Submicron VHSIC program. Specifications for the chip-level ETM-bus[13] and the backplane Test and Maintenance bus[14] (TM-bus) are complete and have been turned over to the DoD for change control and distribution.

A diagram of a Honeywell module or board from a test and maintenance perspective is shown in Figure 1. A module-level Test and Maintenance Processor (TMP) uses commands and data received over the TM-bus to control multiple ETM-buses. The ETM-bus transfers TMP instructions and test data to the TMC function block, which is an ETM-bus slave and is implemented on every VLSI chip in the module. This allows

1987 International Test Conference Proceedings
CH2347-2/87/0000/0964$01.00 © 1987 IEEE
Integration of Test with Design and Manufacturing
1-3 Sep. 1987

for standardized control and monitoring of test and maintenance operations in the functional chip circuitry (labeled "application" in Figure 1).

Before describing the TMC in detail, it is necessary to introduce the other features of the chip which affect TMC operation: the ETM-bus, the chip application, and the clock control logic.

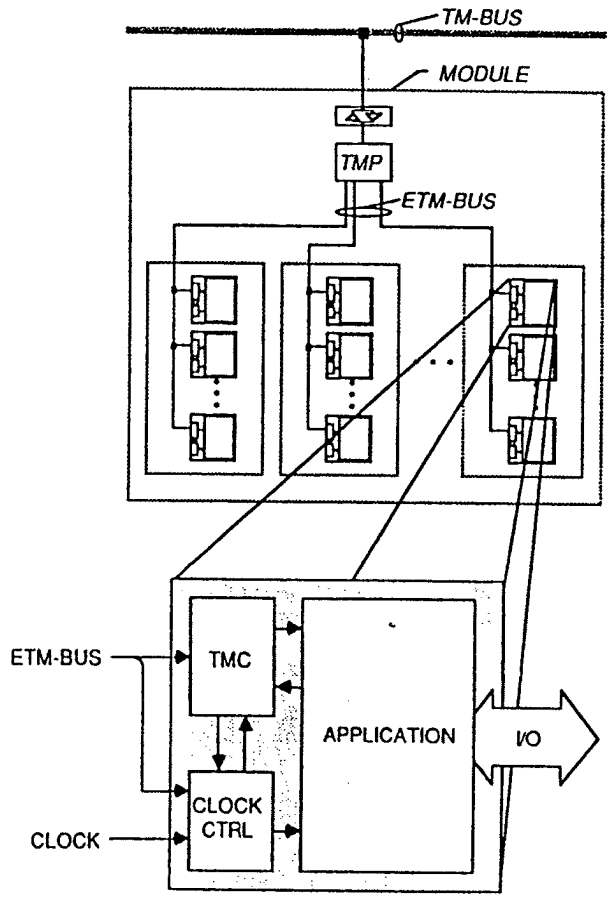
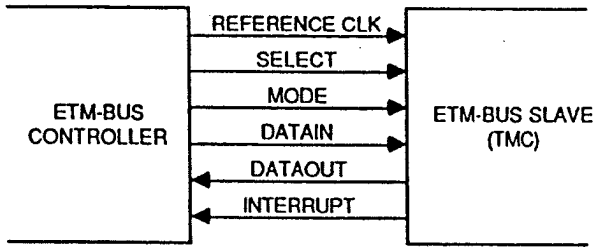


Figure 1: Honeywell's Testable Module

low), as its name implies, selects the appropriate chip in the module for either instruction/status or scan operation, as determined by the MODE signal. During instruction/status operation, the TMC on the selected chip receives a 17-bit serial instruction (16 bits data, 1 bit odd parity) over the ETM-bus DATAIN signal at the same time that it sends status back to the TMP via DATAOUT. When selected for scan operation, the TMC uses the serial data signals to route data to the appropriate scan path on the chip as it sends the previous contents of the scan path to the bus controller. When deselected (SELECT=1), the TMC executes the previously-received instruction.



SELECT	– Bus control signal; signifies data transfer operation
MODE	– Bus control signal; signifies type of operation
DATAIN	– Serial data signal; transfers commands and scan data
DATAOUT	– Serial data signal; supports tri-state operation
INTERRUPT	– Interrupt signal; supports open-collector operation
REF CLK	– Free-running clock signal; 6.25MHz maximum

Figure 2: ETM-bus Signals

SELECT	MODE	Operation
0	0	Instruction/Status
0	1	Scan
1	X	Execute/Idle

Figure 3: ETM-bus Operation

The ETM-bus

Development of the ETM-bus began after the protocol of the multi-drop TM-bus was deemed too complex to be interpreted by an interface implemented as a function block on the typical VLSI chip. The ETM-bus provides a chip-level communication path for test and maintenance information, allowing for a reduced number of drops on the TM-bus.

The ETM-bus consists of the six signals (all TTL-compatible) shown in Figure 2. The state of the two bus control signals, SELECT and MODE, determines the type of operation taking place on the ETM-bus as defined in Figure 3. SELECT (active

For both Instruction/Status and Scan operations, SELECT and MODE are asserted one clock cycle before the bus controller places the least-significant bit of data on the bus, and are released at the same time that the controller places the most-significant data bit on the bus. Figure 4 shows the Instruction/Status data transfer operation during which 17 bits are shifted into each selected TMC. The Scan data transfer operation occurs in the same manner except that MODE is high and there is no limit on the number of bits that may be transferred into each selected TMC. All data transfer operations on the ETM-bus are synchronous with the falling edge of REFERENCE CLK.

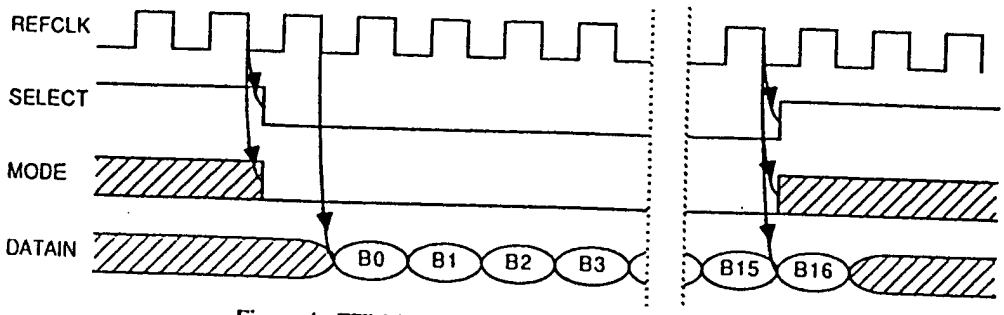


Figure 4: ETM-bus Instruction/Status Operation

The ETM-bus supports both ring and star configurations as shown in Figures 5a and 5b. In the ring configuration, the **DATAIN** signal of the first TMC and the **DATAOUT** signal of the last TMC in the ring are connected to the ETM-bus controller, and the **DATAIN** signal of each remaining TMC is connected to the **DATAOUT** signal of the previous TMC in the ring. All TMCs are simultaneously selected for data transfer operations. Thus, during an Instruction/Status data transfer operation, n consecutive instructions must be shifted ($17 \cdot n$ bits) by the controller for n chips in the ring. In the star configuration, all **DATAIN** and **DATAOUT** signals are connected to the ETM-bus controller and the TMCs are individually selected for Instruction/Status or Scan operation by means of multiple **SELECT** signals.

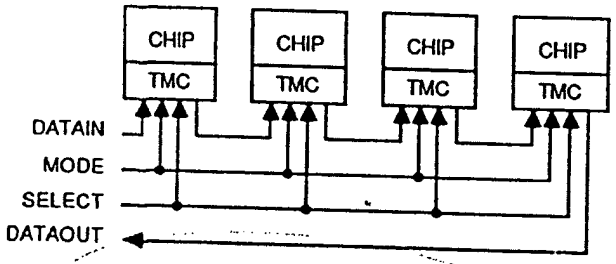


Figure 5a: ETM-bus Ring Configuration

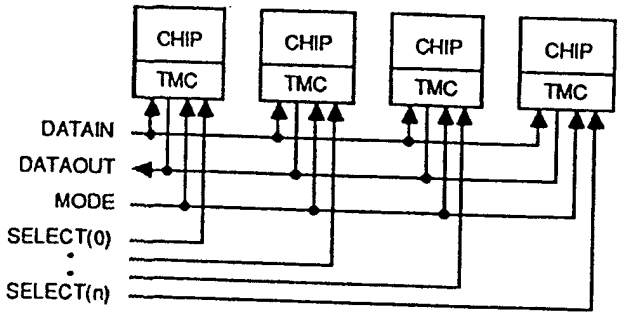


Figure 5b: ETM-bus Star Configuration

The Chip Application

The TMC described in this paper was designed with the assumption that the chip application registers are implemented as reconfigurable serial shift registers supporting a minimum of three configurations: serial scan, reset, and functional (parallel load). Several register designs which support these configurations have been described.[1,8] The TMC uses the decoded ETM-bus state (instruction/status, scan, or execute/idle) and the instructions received from the bus controller to configure the chip application registers. By controlling the registers, the TMC places the chip application into one of the following modes of operation:

- **Functional mode.** The TMC configures the application registers as parallel load registers to provide free-running, functional operation. When the TMC is in its initialized (power-up) state, it places the application into the functional mode, eliminating the need for the bus controller to initiate functional operation by sending an ETM-bus instruction.

- **Debug mode.** The TMC configures the application registers as parallel load registers to provide single- or multiple-step functional operation for chip debug purposes. The length of this operation is controlled by the bus controller via the ETM-bus **SELECT** signal.

- **Serial scan mode.** The TMC uses this mode to gain serial access to all registers on the chip for chip debug and structural verification purposes by configuring the application registers into one or more serial scan paths.

- **Reset mode.** The TMC initializes the chip application or sections of the chip application by synchronously resetting the registers in the selected scan path(s).

- **Built-in Self-test (BIST) mode.** In this mode, the TMC uses a variation of the LOCST method[7] to control built-in self-test of the application. A maximum-length linear feedback shift register (LFSR) within the TMC is configured as a serial test pattern generator (TPG) to shift pseudo-random test patterns into the chip scan paths, and a parallel signature analysis register (PSA) in the TMC is used to simultaneously compact the outputs of all of the scan paths. The length and number of pseudo-random test patterns applied to the application is determined by the bus controller via the ETM-bus control signals.

The Clock Control

Because all ETM-bus data transfer operations are synchronous with the ETM-bus clock, **REFERENCE CLK**, clock switching logic must be included in the design if the application registers are to receive scan data directly from the ETM-bus.

The clock control logic provides accurate switching (no partial clock pulses) between the system clock, used by the application during functional operation, and the ETM-bus REFERENCE CLK, used by the application during test operations such as debug, reset, serial scan, and BIST.

TMC ARCHITECTURE

The TMC, shown in Figure 6, consists of three basic registers (status, transfer, and command), command decode logic, control logic, and parity logic. The TMC has been implemented on an engineering workstation using approximately 500 logic gates (non-specific technology) which makes it suitable for implementation on chips of 10,000 gates or more. Size efficiency was achieved by designing the TMC registers to perform multiple tasks. Since the TMC was designed to be implemented on a serial scan-intensive design, the hardware required by the TMC to control the application registers remains constant as the size of the application changes. Thus the hardware overhead associated with the TMC decreases as the size of the application increases.

The following sections describe the operation of each of the TMC's major function blocks.

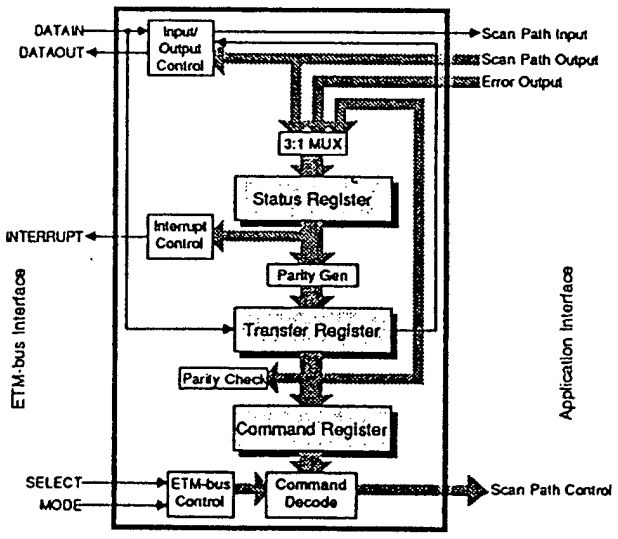


Figure 6: TMC Architecture

enters an intermediate state (Load Command) for one clock cycle during which the TMC checks the parity of the received instruction and loads the instruction into the appropriate register if the parity is correct. If the parity is incorrect, the instruction is ignored. When the machine is in the Scan Data state, the TMC places the application into one of the test modes of operation (debug, reset, serial scan, or BIST), depending on the contents of the command register. When SELECT goes high, the machine enters the BIST state for one clock cycle. This state is used during BIST operation to configure the application registers in the parallel load mode for a single clock cycle.

According to the ETM-bus specification, SELECT must remain high for at least three clock cycles following an Instruction/Status operation and for at least two clock cycles following a Scan operation. This means that the ETM-bus control state machine will stay in the Execute state for at least two cycles following the Load Instruction state and for at least one cycle following the BIST state. This provides the TMC with ample time to set up application register control signals after receiving a new instruction from the ETM-bus controller and before the registers receive the next clock. An additional state could be inserted in the state machine between the Load Instruction and Execute states for the purpose of pipelining decoded instructions.

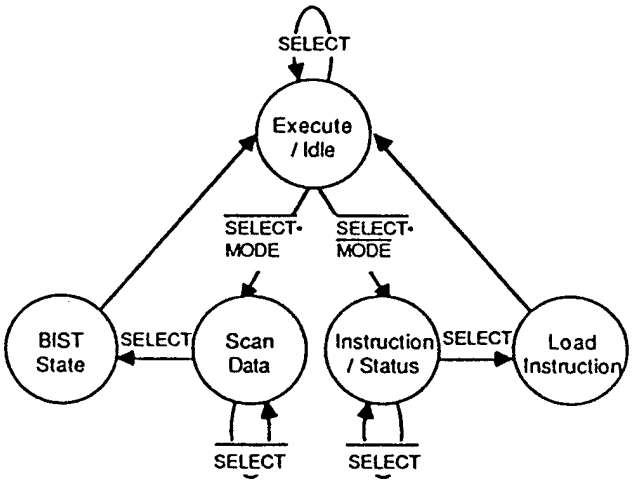


Figure 7: ETM-bus Control State Diagram

ETM-bus Control

The ETM-bus control block is a five-state state machine which monitors the ETM-bus control signals (SELECT and MODE). The state diagram for this machine is shown in Figure 7. The state machine is initialized to the Execute state. When SELECT is asserted, the machine changes to either the Scan Data state or the Instruction/Status state, depending on the value of the MODE signal. When the machine is in the Instruction/Status state, the TMC transfer register is configured as a shift register, and a new instruction is shifted in on DATAIN as the contents of the transfer register are shifted out on DATAOUT. When SELECT goes high, the machine

Transfer Register

The transfer register is a 17-bit register which receives serial data from the ETM-bus for transfer to either the command register or the status register. The transfer register is the only TMC register that is directly connected to the ETM-bus serial data signals. The bus controller loads data into the status and command registers indirectly through the transfer register. Two of the bits in the 17-bit word that is shifted into the transfer register are decoded as shown in Figure 8 to determine the word's destination.

Bits	Destination
11	Command Register
10	Status Register
01	Transfer Register
00	NOOP command

Figure 8: ETM-bus Instruction Destination

When the NOOP (NO OPeration) command is shifted into the transfer register, the command and status register contents are not changed. The NOOP command is particularly useful when the TMC is in a ring configuration of the ETM-bus. If only one TMC in the ring requires an instruction change, the other TMCs receive NOOP commands, relieving the ETM-bus controller of the task of storing all of the current TMC instructions. The NOOP command also allows the bus controller to shift out the contents of the status register (via the transfer register) at any time during functional mode without disturbing the application operation.

The transfer register destination option of the ETM-bus instruction allows the bus controller to initialize the transfer register with a test pattern "seed" prior to beginning BIST operation. The transfer register must be the last register initialized prior to beginning BIST operation since any subsequent instruction transfers on the ETM-bus would destroy the TPG seed. During BIST operation, the transfer register is configured as a maximum-length LFSR and is used to shift the same serial, pseudo-random test pattern into all chip application scan paths. During this mode, the transfer register cycles through 2^{17}-1 states before returning to its original state. Since 2^{17}-1 is a prime number, there is no danger of cycling the same test pattern into a scan path during a single BIST sequence unless the scan path is 2^{17}-1 bits long. Following BIST operation, the contents of both the transfer register and the status register (via the transfer register) are shifted out for verification of their final states.

Status Register

The TMC contains a 16-bit status register which, during functional operation, is used to store asserted error signals from concurrent error detectors on the chip application and within the TMC. The status register is clocked by the application clock so error signals are monitored at functional speed. The use of synchronization logic between the status register and the transfer register is avoided by requiring that the application clock and the ETM-bus clock be synchronous.

During BIST operation, the status register is first loaded with an initial signature "seed." Since two bits of the incoming ETM-bus instruction determine the destination of the instruction, two bits of the status register are always initialized to the same value. When BIST operation begins, the status register is configured as a parallel signature analysis register and compacts the test results shifted out of the application scan paths. The final signature is then loaded into the transfer register and shifted out to the bus controller for verification.

Interrupt Control

The interrupt control logic monitors the status register during functional operation and asserts the ETM-bus **INTERRUPT** signal when an error condition is recorded. The interrupt control logic also monitors the parity checker on the TMC to alert the ETM-bus controller when an instruction with incorrect parity has been sent to the TMC. The control logic continues to assert **INTERRUPT** until the bus controller services the error condition, either by clearing the status register or sending an error-free instruction to the TMC. **INTERRUPT** may be enabled or disabled upon instruction from the bus controller, allowing the controller to ignore interrupts from designated TMCs in the module.

According to the ETM-bus specification, **INTERRUPT** may be asserted asynchronous to the ETM-bus clock, **REFERENCE CLK**. This allows **INTERRUPT** to be asserted at system speed for quicker servicing of application errors.

Command Register

The 14-bit command register is used to store application DFT and fault tolerance control information that must remain stable while new instructions are shifted into the TMC during Instruction/Status operation. Application DFT control signals might include register configuration control, register enables, scan path address, test point control, circuit partition control, etc. Typical fault tolerance control information might include chip configuration control signals, error correction block enable signals, or output buffer disable signals. The bit definition of the incoming instruction is not specified in the ETM-bus specification which allows for flexibility in the design of both the test interface circuitry and the application DFT circuitry.

The size of the TMC command register is limited by the length of the ETM-bus instruction (17 bits minus 1 bit parity) and the number of instruction destinations (2 bits are decoded to determine instruction destination). The following command register control bits are used in the TMC:

- Chip operation instruction (3)
- Scan path address (4)
- Output buffer disable control (2)
- Boundary register control (1)
- Status register multiplexer control (2)
- ETM-bus INTERRUPT enable (1)

The chip operation instruction bits specify whether the application is in the functional, debug, reset, serial scan, or BIST mode of operation. The scan path address bits allow the following choices (13 total) for addressing application scan paths: scan path 0, scan path 1, . . . , scan path 7, input section of boundary scan path, output section of boundary scan path, full boundary scan path, bypass bit (described in the next section), all scan paths. The output buffer disable control signals are used to disable chip outputs when a severe error condition is recorded in the status register. Output buffers are also disabled when shifting the boundary scan path to eliminate switching of primary chip output signals. The boundary register control signal is used to activate any non-functional register bits on the boundary scan path during test operations.

Input/Output Control Logic

The input/output control logic selects which scan path output is shifted onto the ETM-bus **DATAOUT** signal. It also determines the input to the scan paths. During serial scan operation, the ETM-bus **DATAIN** signal is the input to the scan paths, and during BIST operation, the transfer register output is the input to the scan paths.

The input/output control logic also contains an addressable, single-bit scan path called the bypass bit which allows for shorter serial access to individual chips in the ETM-bus ring configuration as shown in Figure 9.

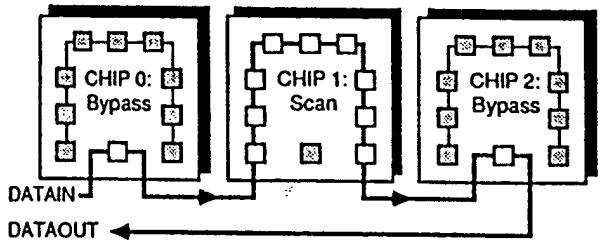


Figure 9: Access to a single scan path in the ring

TMC OPERATION

The chip operation instruction bits in the command register and the state of the ETM-bus control state machine determine the operation of the chip application registers according to the table in Figure 10. The action of the TMC during these different modes of operation is described in the following sections.

ETM-bus Control	Chip Operation Instruction	Application Operation
Execute/Idle OR Instruction/Status OR Load Instruction	Functional	Functional Operation
	Debug	Registers Disabled
	Reset	Registers Disabled
	Serial Scan	Registers Disabled
	BIST	Registers Disabled
Scan Data	Functional	Functional Operation
	Debug	Functional Operation
	Reset	Registers Reset
	Serial Scan	Shift selected scan path
	BIST	Shift selected scan path
BIST State	Functional	Functional Operation
	Debug	Registers Disabled
	Reset	Registers Disabled
	Serial Scan	Registers Disabled
	BIST	Functional Operation

Figure 10: Application Operation

Functional Operation

When the TMC is initialized on power-up, the command register contains the "Functional" instruction. The application remains in the functional mode, regardless of any changes in the ETM-bus control signals, until a new chip operation instruction is loaded into the command register. Any faults on the ETM-bus control signals (SELECT, MODE) will not take the application out of functional operation. If the ETM-bus state machine goes into the Scan Data state, the application remains in the functional mode and the bypass bit is shifted to allow for data transfer operations on the ETM-bus. The bus controller can also shift new instructions into the TMC and read the status register (Instruction/Status mode) without disturbing the application's functional operation, assuming that the chip operation instruction does not change.

Debug Operation

When the ETM-bus state machine is in the Scan Data state and the debug instruction is in the command register, the TMC configures the application registers the same as during functional operation and shifts the bypass bit to allow for data transfer operations on the ETM-bus. When SELECT is released, the ETM-bus state machine goes into the Execute/Idle state and the application registers are disabled. Thus, by asserting the SELECT signal for *n* clock cycles, the controller enables the application registers for *n* clock cycles, providing single- or multiple-step functional operation.

Reset Operation

During reset operation, the selected scan path(s) on the chip are synchronously initialized for a specific number of clock cycles. As with debug operation, reset operation occurs during the Scan Data state and its duration is controlled by the ETM-bus SELECT signal. The bypass bit is also shifted while application registers are being initialized.

Serial Scan Operation

This operation is used to scan data, either structural test patterns or a functional chip state, into the addressed scan paths. If inverting shift paths are implemented, the paths must have an even number of inversions so that in ETM-bus ring configurations, data intended for a mid-ring chip is not modified (inverted) by the previous chips in the ring.

Built-In Self-Test Operation

This operation occurs during the Scan Data state of the ETM-bus state machine when the "BIST" instruction is in the command register. Serial, pseudo-random test patterns from the transfer register are shifted into all scan paths as the scan path outputs are compacted in the status register (Figure 11a). The bypass bit is simultaneously shifted as a scan path on the ETM-bus serial data signals. When SELECT is released, the ETM-bus state machine goes into the BIST state for one clock cycle and the TMC configures the chip registers in the functional mode. This allows the test patterns to propagate through combinational logic and into downstream registers (Figure 11b). Following a single Execute state, during which the application registers are disabled, the ETM-bus state machine may be returned to the Scan Data state, during which the test results are compacted in the status register as new test patterns are shifted into the application registers.

Both the status register and the transfer register may be read by the bus controller for verification of the intermediate or final BIST signatures or test patterns.

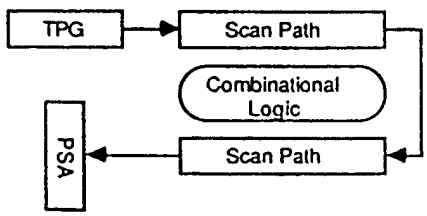


Figure 11a: BIST Operation - Test Pattern Loaded, Test Results Compacted

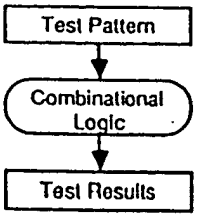


Figure 11b: BIST Operation - Test Pattern Applied

Structural Test

External, non-ETM-bus control signals are used during the structural test process (not shown in Figure 10) to override the TMC signals that control chip application registers. The external control signals, which are multiplexed with application input signals, are also used to configure the registers in the TMC into a single scan path so that computer-generated structural test patterns can be applied to both the TMC and the chip application logic via the serial scan paths. The ETM-bus serial data signals are used to access the scan paths. This structural test configuration allows for high fault coverage of the TMC and the TMC/application interface logic during wafer-level testing.

Master Reset

The TMC uses a non-ETM-bus master reset signal to synchronously initialize all of the TMC and chip application registers. When all of the TMC registers are reset, the TMC places the chip application into the functional mode.

ENHANCEMENTS

The TMC can be easily enhanced to provide control of more complex, chip-specific DFT techniques, yet still remain compatible with simpler TMC designs. These enhancements may become necessary for VLSI chips which contain more complicated, embedded logic such as RAM, ROM, or register files. The enhancements may also become feasible from a hardware overhead perspective if the TMC is implemented on larger VLSI chips.

The TMC architecture allows for easy addition of various command and data registers that may be loaded and monitored by the bus controller in the same manner as the status register. For example, a secondary command register containing register file address signals for embedded register files could be added as shown in Figure 12. Also, a clock counter could be added to control high-speed chip operations such as application debug at system speed or parallel BIST through the use of chip application BILBO[8] (Built-In Logic Block Observation) registers as described by Hudson.[9]

The addition of writeable registers to the TMC requires extra control bits in the ETM-bus instruction to determine the destination of the instruction. Since the length of the ETM-bus instruction is specified as 17 bits, fewer bits are then available for chip testability control information in the incoming instruction.

It may also be desirable to make all of the TMC registers readable by the bus controller to improve the visibility of the TMC. This is done by adding a multiplexer, which is controlled by the command register, upstream of the transfer register and the parity generation block as shown in Figure 12.

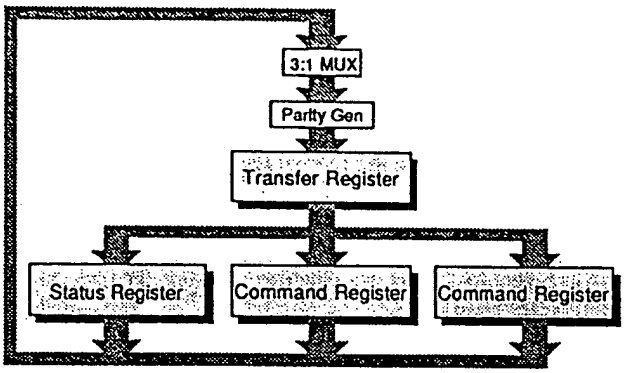


Figure 12: Enhanced TMC with Visible Registers

CONCLUSION

The Test and Maintenance Control (TMC) function block described in this paper provides an interface to the standard VHSIC ETM-bus to form the basis of a structured system testability hierarchy by controlling and monitoring chip testability techniques, built-in self-test operation, and fault monitoring circuitry. The TMC is intended to be implemented as a function block on VLSI chip designs which also make use of advanced, structured DFT techniques such as serial scan and boundary scan. The TMC architecture described in this paper can be easily modified and enhanced to control chips with a variety of testability techniques and has been customized for implementation on several Honeywell VLSI chip designs.

ACKNOWLEDGEMENTS

Some of the work reported in this paper was performed under Air Force Wright Aeronautics Laboratory contract number F33615-84-C-1500.

The author wishes to thank Steve Rasset for his contributions to the modelling and verification of the TMC design.

REFERENCES

[1] Homer W. Miller, "'Design for Test' Via Standardized Design and Display Techniques," *Electronics Test*, pp. 108-115, October 1983.

[2] E.B. Eichelberger and T.W. Williams, "A Logic Design Structure for LSI Testability," *Proceedings 14th Design Automation Conference*, pp. 462-468, 1977.

[3] J.H. Stewart, "Application of Scan/Set for Error Detection and Diagnostics," *IEEE Semiconductor Test Conference Proceedings*, pp. 152-158, 1978.

[4] P. Goel and M.T. McMahon, "Electronic Chip-In-Place Test," *IEEE Test Conference Proceedings*, pp. 83-90, 1982.

[5] John J. Zasio, "Shifting Away From Probes For Wafer Test," COMPCON, pp. 395-398, 1983.

[6] F.P.M. Beenker, "Systematic and Structured Methods for Digital Board Testing," *IEEE International Test Conference Proceedings*, pp. 380-385, 1985.

[7] Johnny J. LeBlanc, "LOCST: A Built-In Self-Test Technique," *IEEE Design and Test*, pp. 45-52, November 1984.

[8] B. Koenemann, J. Mucha, G. Zwiehoff, "Built-In Logic Block Observation Techniques," *IEEE Test Conference Proceedings*, pp. 37-41, 1979.

[9] Charles L. Hudson, Jr. and Gary D. Peterson, "Parallel Self-Test with Pseudo-Random Test Patterns," *IEEE International Test Conference Proceedings*, 1987.

[10] Jon Turino, "A Totally Universal Reset, Initialization (and) Nodal Observation Circuit," *IEEE International Test Conference Proceedings*, pp. 878-883, 1984.

[11] Michael J.Y. Williams and James B. Angell, "Enhancing Testability of Large-Scale Integrated Circuits via Test Points and Additional Logic," *IEEE Transactions on Computers*, Vol. C-22, No. 1, pp. 46-60, January 1973.

[12] V. Blaschke, W. Budde, A. Hunger, "Modular Built-In Testprocessor for Future VLSI-Chips," *Proceedings of EUROCON '86*, pp. 470-475, 1986.

[13] *VHSIC Phase 2 Interoperability Standards ETM-bus Specification*, Version 2.0, December 31, 1986.

[14] *VHSIC Phase 2 Interoperability Standards TM-bus Specification*, Version 2.0, December 31, 1986.